

Computation of Sensitivity Derivatives of Navier-Stokes Equations using Complex Variables

By

Veer N. Vatsa

NASA Langley Research Center, Hampton, VA 23681

Mail Stop 128, email: v.n.vatsa@larc.nasa.gov

Introduction

Accurate computation of sensitivity derivatives is becoming an important item in Computational Fluid Dynamics (CFD) because of recent emphasis on using nonlinear CFD methods in aerodynamic design, optimization, stability and control related problems. Several techniques are available to compute gradients or sensitivity derivatives of desired flow quantities or cost functions with respect to selected independent (design) variables. Perhaps the most common and oldest method is to use straightforward finite-differences for the evaluation of sensitivity derivatives. Although very simple, this method is prone to errors associated with choice of step sizes and can be cumbersome for geometric variables. The cost per design variable for computing sensitivity derivatives with central differencing is at least equal to the cost of three full analyses, but is usually much larger in practice due to difficulty in choosing step sizes. Another approach gaining popularity is the use of Automatic Differentiation software (such as ADIFOR) to process the source code, which in turn can be used to evaluate the sensitivity derivatives of preselected functions with respect to chosen design variables. In principle, this approach is also very straightforward and quite promising. The main drawback is the large memory requirement because memory use increases linearly with the number of design variables. ADIFOR software can also be cumbersome.

some for large CFD codes and has not yet reached a full maturity level for production codes, especially in parallel computing environments.

Another viable methodology for computing sensitivity derivatives is based on the adjoint approach. This method has received a lot of publicity in recent years due to the pioneering work of Jameson and colleagues, who had focused their attention primarily on inviscid flows [1]. A few applications of this approach to Navier-Stokes equations have also become available in the last couple of years [2-4]. The adjoint approach offers great savings and lower memory penalty for a large number of design variables. The major disadvantage of this method is that a different set of adjoint equations must be derived and solved for each different cost function. The overall process of deriving the adjoint equations to achieve compatibility with the finite-difference equations and the corresponding boundary conditions is tedious and time consuming.

An alternate approach, based on the use of complex variable expansions for computing sensitivity derivatives, was suggested by Lyness [5] and Lyness and Moler [6]. For some unknown reasons, such as the inability of compilers to deal effectively with complex arithmetic, this technique has not been exploited much until Squire and Trapp [7] revived it. Anderson and colleagues [8] have recently made use of complex variables to evaluate sensitivity derivatives with an unstructured-grid Navier-Stokes flow solver. This technique is quite straightforward to apply and produces accurate sensitivity derivatives without suffering from step size related numerical problems, as has been demonstrated by Anderson et. al [8].

This paper discusses the application of complex variables for evaluation of sensitivity derivatives

used in conjunction with the multiblock Navier-Stokes code TLNS3D-MB [9]. Several examples are presented to establish the accuracy and efficiency of this methodology. Some possible uses of the resulting complex variable code are demonstrated, including the computation of force polars and stability derivatives for aerodynamic problems of current interest.

Evaluation of Sensitivity Derivatives

The Taylor series expansion of a function can be written in terms of complex variables as follows:

$$f(x + ih) = f(x) + ih \frac{df}{dx} - \frac{h^2}{2} \frac{d^2 f}{dx^2} - \frac{ih^3}{6} \frac{d^3 f}{dx^3} + \text{hot}$$

Separating the real and imaginary parts of this equation, we get

$$\text{Im}\{f(x + ih)\} = h \frac{df}{dx} - \frac{h^3}{6} \frac{d^3 f}{dx^3} + \text{hot}$$

$$\text{Re}\{f(x + ih)\} = f(x) - \frac{h^2}{2} \frac{d^2 f}{dx^2} + \text{hot}$$

Neglecting the higher order terms of the expansion, the gradient (first derivative) of the function f with respect to x can be written as

$$\frac{df}{dx} = \frac{\text{Im}\{f(x + ih)\}}{h}$$

The second derivative can be expressed in terms of the real part of the function in a similar manner. Notice that the first derivative or the sensitivity derivative given by the above equation in terms of complex variables avoids the subtraction of nearly equal numbers, as opposed to the standard finite-differencing methods, and therefore does not exhibit the accuracy problems associated with small step sizes.

Conversion Procedure for Complex Variables

The procedure for converting an existing Fortran computer code to complex variables is relatively simple and straightforward. The computer code used in this exercise was the multiblock Navier-Stokes code known as TLNS3D-MB [9]. It is a general purpose CFD code which can be used for solving turbulent, viscous flows over complex configurations. This code uses a multigrid-based Runge-Kutta time-stepping scheme to obtain steady-state solutions in an efficient manner. A coarse-grain parallelism exploiting the multiblock data structure is incorporated in this code which consists of over 110 subroutines in its latest version. The conversion procedure is summarized in the following steps.

1. Insert the statement “Implicit Complex (a-h, o-z)” immediately after the program, subroutine or function statement in each Fortran module.
2. Create a module ccomplex.f, which contains the complex equivalent of the intrinsic routines, such as max, min, abs, mod, dim, tanh,

atanh, cos, acos

3. In each program unit, replace reference to the above mentioned routines with their complex counterparts; e.g. replace “max” with “ccmax”
(Note: ccmax and ccmin only support 2 arguments)
4. Replace “logical if” statements in the code involving floating point variables in the following manner:

old: if(x.gt.y) then

new: if(real(x).gt.real(y)) then

(similar procedure for .ge., .lt. and .le. statements)
5. Modify the I/O statements so that read and write statements correspond to correct data types.
6. In the MPI_SEND and MPI_RECV calls, replace the floating point variable references by the corresponding complex variable references.

e.g., replace MPI_Double_Precision with MPI_Double_Complex.

Similar procedure could be applied to any other computer code. It may be necessary to take care of specific intrinsic calls in a manner similar to the procedure described in step 2 to make these calls compatible with complex arguments. Once the steps outlined above have been carried out, and user-created bugs resolved, the code is almost ready to run. Another detail required is the identification of independent (or design) variables. Say for example that we are interested in computing the sensitivity derivatives with respect to the freestream Mach number, M . Then we need to add a small imaginary component (h) to the variable M in the beginning of the code, after the initial value of M has been set. The derivative of any of the computed variables (including integrated

forces or cost functions) with respect to M is then given by the imaginary part of that variable divided by h .

Although the complex variable approach for computing sensitivity derivatives offers many advantages, it also has a few disadvantages. To start with, the complex version of the code requires nearly double the memory of the original code and takes approximately 2.5 - 3 times more computer time due to the complex arithmetic. In addition, because the conversion to complex variables requires human intervention, the complex variable approach raises the possibility of introducing user-induced errors.

Results and Discussion

The first test case presented here is for inviscid, transonic flow over the RAE 2822 airfoil. The baseline angle of attack was chosen as 6 degrees at a free stream Mach number of 0.75. A C-grid consisting of 129×49 points was used in these computations. The sensitivity derivatives with respect to free stream Mach number were computed first by the standard finite-difference method. Based on the numerical experimentation, a step size of 0.0001 in Mach number produced accurate derivatives of force coefficient. When the complex variable approach was employed, very little sensitivity was observed in the values of computed derivatives, and hence only the computations with the smaller value of step size (0.00001) are reported here in Table 1.

Table 1: Sensitivity derivatives with Mach number for RAE 2822 Airfoil

	$\frac{\partial C_l}{\partial M}$	$\frac{\partial C_d}{\partial M}$	$\frac{\partial C_m}{\partial M}$
Finite-difference ($h = 0.001$)	0.65	1.16	-2.59
Finite-difference ($h = 0.0001$)	0.3795	1.1099	-2.4101
Complex Variables	0.34683	1.10412	-2.38873

Note that the sensitivity derivatives computed with the complex variable approach are much closer to the finite-difference computations with smaller step size, as expected because the accuracy of finite-difference solutions improves with reduction in step size. Similar comparisons for this case are shown for sensitivity derivatives with respect to the angle of attack in Table 2. The step size used for the finite difference solutions was 0.0001 in order to reduce truncation error in these solutions. The comparisons between these sensitivity derivative computations are quite good. Based on the results so far, we believe that the complex variable method produces accurate and reliable derivatives of force coefficients with respect to free stream Mach number and angle of attack for inviscid transonic flow over the RAE 2822 airfoil.

Table 2: Sensitivity derivatives with angle of attack for RAE 2822 Airfoil

	$\frac{\partial C_l}{\partial \alpha}$	$\frac{\partial C_d}{\partial \alpha}$	$\frac{\partial C_m}{\partial \alpha}$
Finite-Difference	0.0993	0.03425	-0.0396
Complex Variables	0.09928	0.03421	-0.03958

Next we move to more complex flow and geometry and consider the viscous flow over a four-block ONERA M6 wing configuration. The baseline test conditions were selected as follows: Mach number of 0.84, angle of attack of 3.06, and a Reynolds number (based on mean aerodynamic chord) of 11.7 million. The grid consisted of approximately 312,000 mesh points. The Spalart-Allmaras turbulence model [10] was used in these computations. The sensitivity derivatives with respect to both the angle of attack and freestream Mach number are presented in Table 3. The sensitivity derivatives computed with the complex variables method are in excellent agreement with the finite difference solutions for this test case, verifying the applicability of the complex variable approach to realistic, three-dimensional viscous, turbulent flows at transonic speeds.

Table 3: Sensitivity Derivatives for ONERA M6 Wing

	$\frac{\partial C_l}{\partial \alpha}$	$\frac{\partial C_d}{\partial \alpha}$	$\frac{\partial C_m}{\partial \alpha}$	$\frac{\partial C_l}{\partial M}$	$\frac{\partial C_d}{\partial M}$	$\frac{\partial C_m}{\partial M}$
Finite-Difference	0.0761	0.0058	-0.0284	0.227	0.118	-0.16
Complex Variables	0.07610	0.00581	-0.02839	0.22731	0.11872	-0.15965

Having established the accuracy of the complex variable approach for computing the sensitivity derivatives of integrated forces, we decided to investigate the possibility of using this approach for reducing the cost of generating force coefficient polars over a range of test conditions. The procedure is as follows. We generate a well converged baseline solution at a given angle of attack and save this solution on a restart file. In order to compute the solution at a different angle of attack (within 1 to 2 degrees of baseline), the baseline solution is read in from the restart file. The real part of this solution is modified as follows:

$$Q(\alpha) = Q(\alpha_{baseline}) + (\alpha - \alpha_{baseline}) \frac{\partial Q}{\partial \alpha}$$

The above equation is applied to all of the flow variables, which provides a much better initial guess at the new angle of attack. Consequently, the solution converges much more quickly (about 50 cycles instead of 300 cycles from cold start) to acceptable levels of convergence. This method was employed to generate the lift curve over the entire angle of attack range shown in Figure 1,

starting with a baseline solution at 1.64 degrees for the NACA 0012 airfoil on a 193×65 grid. Experimental data of Harris, modified with his angle of attack corrections [11] is shown for comparison. Although not shown, the solutions obtained in conventional manner (running about 300 cycles from freestream start) are indistinguishable from the computed solutions shown in this figure.

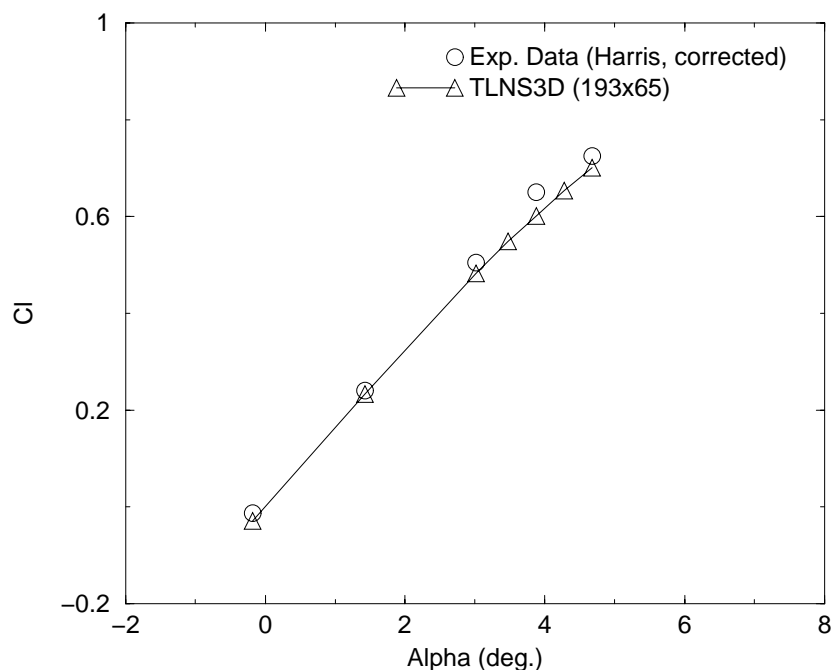


Fig. 1: Lift curve for NACA 0012 Airfoil; $M = 0.7$, $Re = 9$ million

The final test case considered here represents an unconventional aircraft configuration known as the Blended Wing Body (BWB) configuration, which is being evaluated as one of the advanced concepts at NASA. This configuration has very peculiar stability characteristics in flight. At the present time, the stability derivatives for such configurations are obtained primarily from experimental data because conventional linear and inviscid based computational methods cannot accu-

rately predict these characteristics, especially under transonic flight conditions.

A grid consisting of $289 \times 65 \times 61$ points, split evenly into 8 blocks, was used to model this configuration. The Mach number of 0.85 and chord Reynolds number of 25 million were chosen for the computations to correspond to an existing experimental data base [12]. The turbulence model of Spalart and Allmaras chosen for these computations [10].

The CFD results for lift coefficient presented in Figure 2 are in excellent agreement with the experimental data, and accurately capture the break in the lift curve slope at about 2.7 degrees angle of attack. Similar correlation with the experimental data is also seen for the pitching moment polar (Figure 3). It is our understanding that knowledge of the reversal in pitching moment polar is crucial from a stability and control viewpoint, and the present CFD code is shown to predict these trends accurately (Figure 3) for the BWB configuration examined in this study.

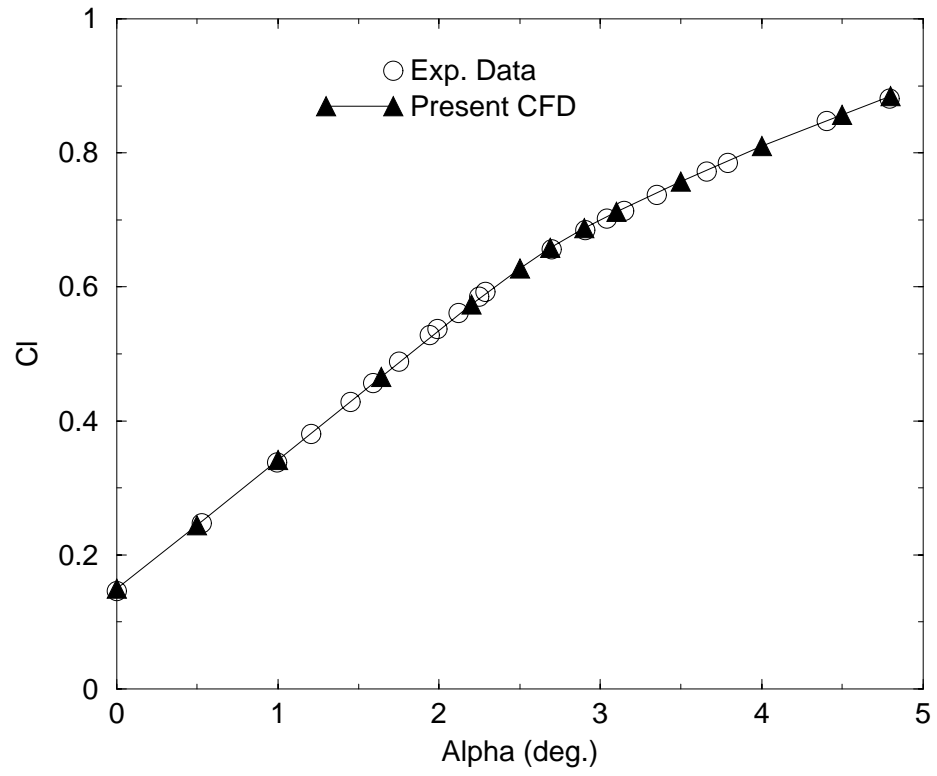


Fig. 2: Lift coefficient comparisons for BWB configuration; $M = 0.85$, $Re = 25$ million

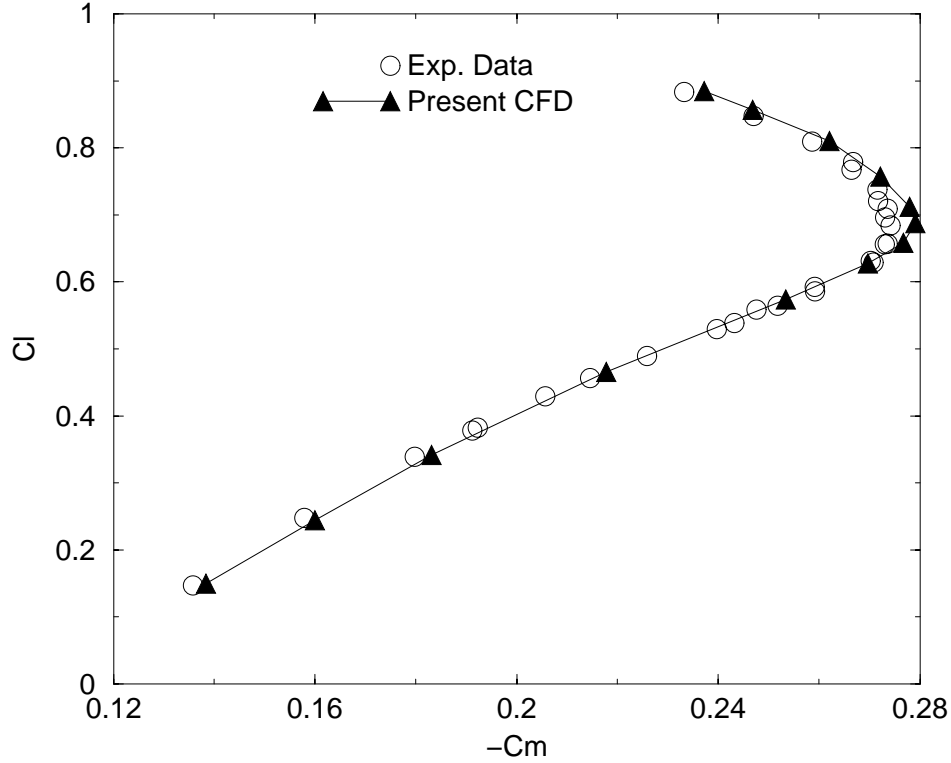


Fig. 3: Pitching moment polar for BWB configuration; $M = 0.85$, $Re = 25$ million

The complex variable approach outlined in this paper can provide accurate values of stability derivatives at all the computational points. The variations of $\frac{\partial C_l}{\partial \alpha}$ and $\frac{\partial C_m}{\partial \alpha}$ with angle of attack are shown in Figures 4 and 5 respectively, where a large change in the slope is observed at approximately 2.2 degrees angle of attack. It is also noted that the slope of the curve for pitching moment derivative reverses sign at approximately 3.0 degrees angle of attack (Figure 5). Such information would be very useful for control purposes, and is difficult to obtain by conventional finite-difference techniques due to the highly nonlinear nature of force coefficient polars for such configurations. A large number of data points with very fine increments in angle of attack (or whatever the

chosen independent variable is) would be required in order to accurately compute the stability derivatives, resulting in significant computational costs. On the other hand, experimental data is also hard to generate with regular increments in test conditions and suffers from repeatability issues. To illustrate this point, the derivatives obtained by finite-differencing of experimental data are also plotted on these figures. The overall comparison is quite good except for large scatter in the experimental data, which is possibly caused by the errors associated with repeatability of test conditions. The proposed approach thus offers an efficient, accurate, and practical tool for such applications.

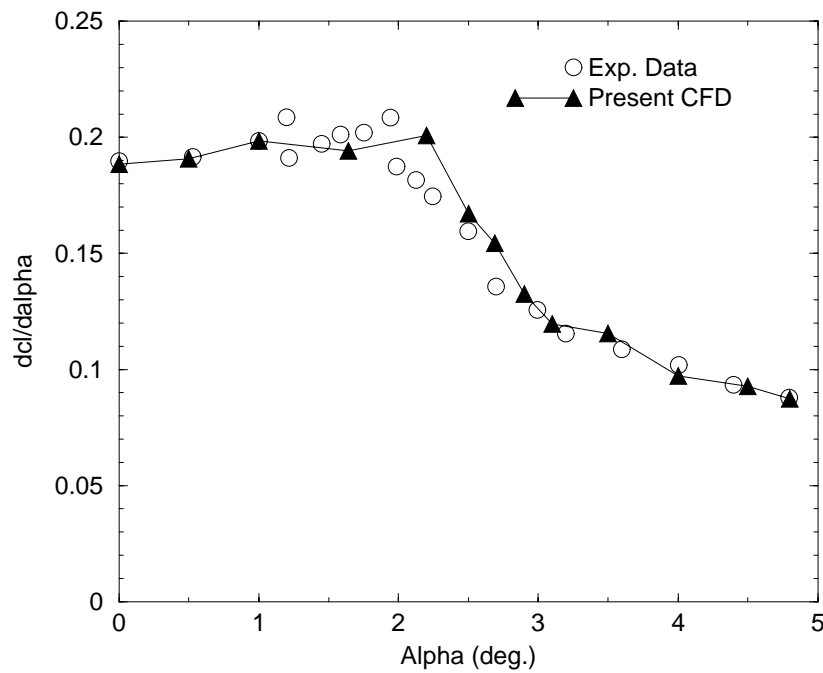


Fig. 4: Variation of lift-coefficient derivative with angle of attack
for BWB configuration; $M = 0.85$, $Re = 25$ million

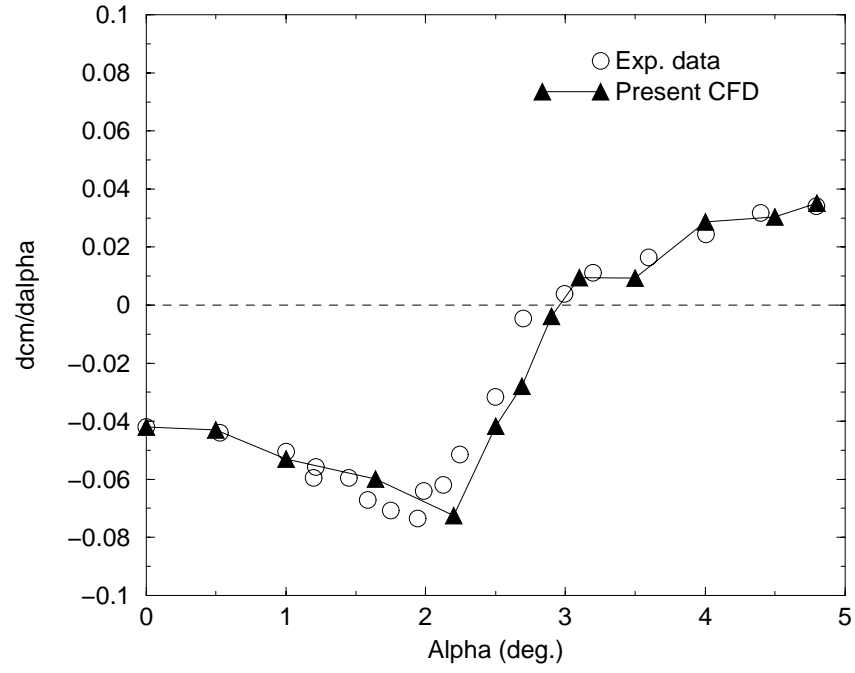


Fig. 5: Variation of pitching moment derivative with angle of attack
for BWB configuration; $M = 0.85$, $Re = 25$ million

Concluding Remarks

A convenient and straightforward procedure using complex variables has been outlined and demonstrated for computing sensitivity derivatives in a multigrid-based multiblock Navier-Stokes code. The accuracy and efficiency of this procedure has been demonstrated through various applications, including several two-dimensional airfoils, a three-dimensional wing, and an advanced Blended Wing Body (BWB) configuration. The proposed method has also been demonstrated to efficiently generate solutions over a large range of test conditions by providing a better starting solution estimated through the use of sensitivity derivatives. For the BWB configuration, computed lift and pitching moments were shown to be in excellent agreement with the experimental data in the transonic flow regime. In addition, the computed stability derivatives indicated crucial pitchup features more clearly than was observed from examination of force and moment curves alone.

Acknowledgments

The author is deeply indebted to Dr. W. K. Anderson of NASA Langley for bringing the complex variable method to his attention and helping with programming issues associated with this approach. The author would also like to express his gratitude to Dr. N. T. Frink of NASA Langley for providing the grid for the Blended Wing Body Configuration.

References

1. Jameson A. Aerodynamic design via control theory. J Sci Comp 1988; 3: 233-260.
2. Anderson WK. Bonhaus DL. Airfoil design on unstructured grids for turbulent flows. AIAA Journal 1999; 37(2):185-191.
3. Anderson WK. Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with continuous adjoint formulation. Computers and Fluids 1999; 28(4-5):443-480.
4. Jameson A. Pierce NA. Martinelli L. Optimum aerodynamic design using the Navier-Stokes equations. AIAA Paper 97-0101, 1997.
5. Lyness JN. Numerical algorithms based on the theory of complex variables. Proc ACM 22nd National conference. Washington DC: Thomas Book Company, 1967; 124-134.
6. Lyness JN. Moler CB. Numerical differentiation of analytic functions. SIAM J Num Anal 1967; 4: 202-210.
7. Squire W. Trapp G. Using complex variables to estimate derivatives of real functions. SIAM Review 1998; 10(1):110-112.
8. Anderson WK. Newman JC. Whitfield DL. Nielson EJ. Sensitivity analysis for the Navier-Stokes equations on unstructured meshes using complex variables. AIAA Paper 99-3294, 1999.
9. Vatsa VN. Sanetrik MD. Parlette EB. Development of a flexible and efficient multigrid-based multiblock flow solver. AIAA Paper 93-0677, 1993.
10. Spalart PR. Allmaras SR. A one-equation turbulence model for aerodynamic flows. AIAA Paper 92-0439, 1991.
11. Harris CD. Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8-foot transonic pressure tunnel. NASA TM 81927, 1981.
12. Pelkman RA. Key findings & conclusions from an NTF wind tunnel test of an initial blended-wing-body concept. CRAD-9402-TR-3985, NASA Contract NAS1-20268, 1998.